MECH 550C/575A: Principles of Mechatronics – Software Engineering

Winter 2020

University of British Columbia

Department of Mechanical Engineering

Course Instructor:

Instructor:	**Dr. Christoph Sielmann (**sielmann@mail.ubc.ca)
Lectures:	Tuesday and Thursday from 2:00 – 3:30PM in MacMillan 260
Website:	Course content available through **UBC Canvas** (http://canvas.ubc.ca)
Contact:	Office hours Tuesday 1:00 – 2:00PM before class in CEME 2208. Please include "MECH550C" in the subject line of any emails.
Lab	MECH 575A is required for this course. Labs are Wed 3PM – 5PM in RH123.

Teaching Assistant(s):

Minsi Sung (minsi.sung@ubc.ca)
MohammadHossein Rahimi (H.Rahimi@ubc.ca)

Course Structure

Lecture slides will be provided 24 hours in advance of the lecture on Canvas. Please be punctual when arriving to class! Assignments are intended to be completed within the designated lab time where TAs will help to provide guidance. The assignments will be distributed through Canvas and will be due midnight on the on the due date. Instructions for submitting assignments will be provided.

Learning Outcomes or Objectives:

This course serves as an introduction to software programming in mechatronics, presented using C# as an example language. Emphasis is placed on software design principles, style, and practical considerations such as commenting, maintenance, and change control. The following lists the relevant units:

- **UNIT 1: Introduction to Programming**
- **UNIT 2: Object Oriented Design**
- **UNIT 3: Data Structures**
- **UNIT 4: Software Engineering: Specifications and Design Principles**

Course Schedule and Topics

The course is designed to front load the more complicated, theory-oriented components of the course. The latter half contains conceptually simpler material, but students are expected to take on a project in addition to the weekly labs.

Date	Activity	Notes
Jan. 7	Introduction	Course introduction and Introduction to Software
Jan. 9	Languages	Overview of programming languages, types, and pros/cons
Jan. 14	OOP #1	Tools and Program Anatomy
Jan. 15	Lab #1	
Jan. 16	OOP #2	Expressions and operators
Jan. 21	OOP #3	Functions and function parameters
Jan. 22	Lab #2	
Jan. 23	OOP #4	Classes, inheritance, polymorphism, and references
Jan. 28	OOP #5	Access modifiers and constants
Jan. 29	Lab #3	
Jan. 30	OOP #6	Tuples, virtual methods, and overloading
Feb. 4	OOP #7	Abstract classes, interfaces
Feb. 5	Lab #4	
Feb. 6	OOP #8	Conditionals and enumerations
Feb. 11	OOP #9	Exceptions
Feb. 12	Lab #5	
Feb. 13	OOP #10	Review
Reading Week		
Feb. 25	Data Structures #1	Arrays, structs, unions, stacks, lists
Feb. 26	Lab #6	
Feb. 27	Data Structures #2	Dictionaries, hashes, JSON, traversal
Mar. 3	Data Structures #3	Collections, sorting, final project introduction
Mar. 4	Lab #7	
Mar. 5	Review	In class review of material
Mar. 10	**In Class Midterm**	
Mar. 12	Version Control	Version control methods
Mar. 17	Style #1	Object Oriented Design style
Mar. 18	Lab #8	
Mar. 19	Style #2	Comments, Linters, Score, and maintainable code
Mar. 24	Software Engineering	Project lifecycle, agile development, dev ops
Mar. 25	Lab #9	
Mar. 26	Special Topics	Random numbers, GUIs
Mar. 31	Data Exchange	Files, REST, XML, CSV, and others
Apr. 1	Lab #10	
Apr. 2	Defensive Programming	Defensive coding and debugging
Apr. 7	Scaling Software	From miniature MCU to massive cluster

Learning Activities

The course activities include

- Lectures held twice a week.
- Ten labs/assignments held weekly throughout the semester. TAs will be available during the labs to provide learning support. See the calendar on Canvas for scheduling information.
- Instructor office hours and TA meetings are also available to supplement learning.

Learning Materials

There are no required textbooks for the course, but useful literature includes:

- Professional C# 6 and .NET Core 1.0 by Christian Nagel

- Massive resources, including full courses available online:
  - https://www.codecademy.com/learn/learn-c-sharp
  - https://www.learncs.org/
  - https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/
  - https://stackify.com/learn-c-sharp-tutorials/
  - https://www.w3schools.com/cs/
  - https://www.guru99.com/c-sharp-tutorial.html
- On object oriented programming style/technique:
  - https://codeburst.io/how-to-do-object-oriented-programming-the-right-way-1339c1a25286
  - https://www.freecodecamp.org/news/object-oriented-programming-concepts-21bb035f7260/

Assessment, Evaluation, and Grading

- 10 assignments (lab work): 50%
- Midterm: 20%
- Term project: 30%

Both MECH 550C and 575A will be assigned the same grade.

Computers will be available in the labs, but students are encouraged to install relevant development tools on their own laptops for easier access and future use. Required software will be posted on the course website.

Deadlines

Assignments are due by 11:59PM on the indicated date (typically two weeks after assignment). Late assignments and lab reports will be docked 15% per day.

Academic Misconduct

The academic enterprise is founded on honesty, civility, and integrity. As members of this enterprise, all students are expected to know, understand, and follow the codes of conduct regarding academic integrity. At the most basic level, this means submitting only original work done by you and acknowledging all sources of information or ideas and attributing them to others as required. This also means you should not cheat, copy, or mislead others about what is your work. Violations of academic integrity (i.e., misconduct) lead to the breakdown of the academic enterprise, and therefore serious consequences arise and harsh sanctions are imposed. For example, incidences of plagiarism or cheating may result in a mark of zero on the assignment or exam and more serious consequences may apply if the matter is referred to the President's Advisory Committee on Student Discipline. Careful records are kept in order to monitor and prevent recurrences.

Academic honesty is a fundamental requirement of your studies. It is your obligation to inform yourself of the applicable standards. More information is available at http://calendar.ubc.ca/vancouver/index.cfm?tree=3,54,111,0].

Policies and Resources to Support Student Success

UBC provides resources to support student learning and to maintain healthy lifestyles but recognizes that sometimes crises arise and so there are additional resources to access including those for survivors of sexual violence. UBC values respect for the person and ideas of all members of the academic community. Harassment and discrimination are not tolerated nor is suppression of academic freedom. UBC provides appropriate accommodation for students with disabilities and for religious and cultural observances. UBC values academic honesty and students are expected to acknowledge the ideas generated by others and to uphold the highest academic standards in all of their actions. Details of the policies and how to access support are available at https://senate.ubc.ca/policies-resources-support-student-success. Mechanical Engineering also has a Student Services Office (students@mech.ubc.ca), located in CEME 2205, where there are staff who can provide support and refer students to the appropriate resources.

Inclusive Environment

The Department of Mechanical Engineering is committed to providing an inclusive learning experience, and affirms the UBC Statement on Respectful Environment (https://www.hr.ubc.ca/respectful-environment/files/UBC-Statement-on-Respectful-Environment-2014.pdf). You are encouraged to contact their instructor should situations arise that are not consistent with this expectation. You are also invited to advise the instructor if you wish to be addressed by or referred to with particular pronouns.

Laboratory Safety

UBC Mechanical Engineering considers safety first, and continuously, in its labs, research, and other activities. Students are expected to engage in safety discussions; to ask questions to ensure they understand safety information; to comply with policies and rules; to maintain a safe workspace; and to report all accidents, incidents, and near misses immediately to their supervisor and to https://cairs.ubc.ca. Students should work with their supervisors to ensure they understand (1) the risks associated with their work and (2) how those risks are controlled.